社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－HPC－107 (6)
2006／7／31

並列TCPストリームのための流量割り当て方式

菅原 豊　　稲葉 真理　　平木 敬

東京大学 大学院情報理工学系研究科 コンピュータ科学専攻
〒113-8656 東京都文京区本郷 7-3-1
Email: {sugawara, mary, hiraki}@is.s.u-tokyo.ac.jp

要旨

複数のTCPストリームを用いてデータ転送を行う場合，転送時間短縮のためにはストリーム間の速度差を抑える必要がある．ストリーム間の速度差を抑えるための既存の方式では，エンドホストのソフトウェアを変更する必要がある．本研究では，エンドホストに変更を加える事無くストリーム間の速度ばらつきを抑える機構，Stream Harmonizerを提案する．また，提案方式を実装し，10ギガビット・イーサネット環境において評価する．評価の結果，提案方式によりストリーム間の速度差を抑制し，最も遅いストリームの速度を改善できる事が分かった．

Flow Assignment Method for Parallel TCP Streams

Yutaka Sugawara　　Mary Inaba　　Kei Hiraki

Department of Computer Science, Graduate School of Information Science and Technology,
University of Tokyo
7-3-1 Hongo Bunkyo-ku Tokyo 113-8656, Japan
Email: {sugawara, mary, hiraki}@is.s.u-tokyo.ac.jp

Abstract

When using multiple TCP streams to transfer data, it is necessary to reduce throughput difference among streams to reduce the data transfer time. To use existing methods to reduce performance difference among streams, it is necessary to modify the software of the end hosts. In this paper, we propose the Stream Harmonizer, a mechanism to reduce performance difference among streams without modification to the end hosts. In addition, we implement the proposed method and evaluate the implementation using 10Gbit Ethernet environment. In the evaluation, we found that the proposed system reduced performance difference among streams, and improved the throughput of the slowest stream.

1　はじめに

近年，長距離広帯域ネットワークを用いてクラスタ間で高速にデータを転送する技術が求められている．例えば，Data-Reservoir [1] では異なる拠点に置かれたクラスタの間でハードディスクのデータを共有するため，高速なデータ転送が必要である．クラスタの各ノード毎にTCP/IPコネクションを張ってデータ転送を行う場合，システム全体としては複数のTCPストリームを用いて転送を行う．このよ

うに協調してデータ転送を行うTCPストリームの組を並列TCPストリームと呼ぶ．本研究では，個々のTCPストリームが送るデータのサイズは同じであるという前提で議論を行う．

並列TCPストリームを用いてデータ転送を行う場合，データ転送時間を短縮するにはTCPストリーム間の速度差を抑える事が重要である．並列TCPストリームでは，システム全体としてのデータ転送時間は最も遅いストリームがデータ転送に要した時間に等しい．したがって，データ転送時間を短縮するには，並列TCPストリーム全体の合計速度が同じならストリーム間の速度差を抑える必要がある．

TCP/IPではネットワーク途中経路での輻輳を防ぐため送信側で流量制御を行う．送信側では，ACKを受け取らずに送るデータ量を輻輳ウィンドウ(Congestion Window, CWND)という値以下に制限する．CWNDを動的に変えるアルゴリズムにはRenoやBIC等がある．これらのアルゴリズムでは，送出したパケットに対するACKを受信する度にCWNDを増やす．また，パケットロス等で一定時間内にACKが届かなかった場合にCWNDを減らす．

並列TCPストリームを用いてデータ転送を行った場合，ストリーム間で速度にばらつきが生じることが知られている[4]．これは，ストリーム間でCWNDにばらつきが生じるためである．図1，図2は，ボトルネックが存在する高遅延ネットワークで16本のTCPストリームを用いて通信を行った場合の，各ストリームの通信速度変化である．輻輳制御方式はBICで，帯域のボトルネックと遅延は本稿で述べる10GbE多目的実験装置を用いて再現した．

図2から，ストリーム間に速度格差が付き，また，一度付いた格差がなかなか縮まらないことが分かる．最も遅いストリームは，他の速いストリームがパケットロスを起こすと同時にパケットロスを起こすため，CWND値が増加しにくい．また，使用可能なボトルネックリンク帯域が他の速いストリームにより占有されているため，CWNDが伸びきる前にパケットロスが発生してCWNDが減少する．

我々は，速すぎるストリームの速度を抑える事によりストリーム間の格差を縮める方式を提案してきた．[4]ではLinuxカーネルに改造を施し，速すぎるストリームの速度を抑えるよう送信側ノード間で調停を行う．また，[5]ではエンドノードからCWNDを入力として受け取り流量を調節するネットワーク装置を用いてストリーム間の速度差を抑える．しかし，これらの方式は各エンドホストのソフトウェアの変更が必要であるという点に限界があった．

本研究では，指定された並列ストリームのストリーム間速度格差を抑えるため，送信側クラスタとWANの中間において平均より速いストリームの送出速度を抑える機構を提案する．既存方式と異なりエンドホストへの変更は不要である．提案する機構をStream Harmonizerと命名する．速いストリームの速度を抑えることにより，速いストリームが他のストリームを巻き添えにしてパケットロスを起こす現象が起きる頻度を減らす．また，遅いストリームのCWNDが増加するための利用可能帯域の空きを確保する．パケットをバッファにためて送出を遅らせることにより速いストリームの送出レートを抑える．制御対象となるTCPストリームはユーザが予め指定する．本稿では，Stream HarmonizerをFPGA搭載の10GbE実験装置TGNLE-1を用いて実装する．また，実装したStream Harmonizerをボトルネックリンクのエミュレータを用いて評価する．

本稿では，まず2章でStream Harmonizerについて述べる．次に3章でStream Harmonizerの実装について述べる．4章では，ボトルネックリンクのエミュレーション装置を用いて実装したStream Harmonizerの評価を行う．5章で関連研究について述べる．最後に6章でまとめを行う．

2 Stream Harmonizer

2.1 速いストリームの送出レート制限

並列ストリーム全体の合計スループットが同じであれば，データ転送時間を短縮するにはストリーム間の速度差を減らす必要がある．そのためには平均より速いストリームの速度を抑える必要がある．

Stream Harmonizerでは，指定された並列ストリームに対し，各ストリームの速度が全ストリームの平均速度を一定以上こえないよう送出レートを抑える．この制御により平均より速いストリームの速度を抑える．パケットをバッファリングして送出時期を遅らせることにより送出レートを制限する．

送出レート上限の決定　時間間隔t_s毎に各ストリームの流量測定と送出量上限の決定を行う．t_sをスケジューリング間隔と呼ぶことにする．並列TCPストリームを構成するストリーム本数をNとする．時

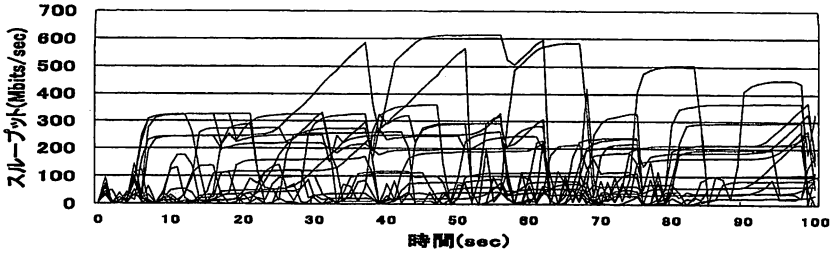

図 1: 並列 TCP ストリームにおける各ストリームのスループット

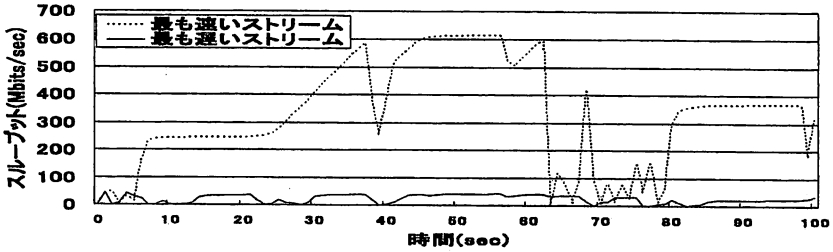

図 2: 並列 TCP ストリームにおけるストリーム間の速度ばらつき

間 t_s の間に Stream Harmonizer が送出した i 番目のストリームのパケット個数が c_i だった場合，次の t_s 時間内に送るパケット数を 1 ストリームあたり $(\sum c_i)/N + \alpha$ 以内に抑える．α は正の定数である．$(\sum c_i)/N$ は 1 ストリームあたりの平均パケット送出数であるため，時間 t_s あたりの各ストリームのパケット送出個数が平均値 $+\alpha$ 以内に抑えられる．

送信ホストへのバックプレッシャ　Stream Harmonizer ではパケットをバッファリングする事により送出レートを抑制している．そのため，もし送信ホストが Stream Harmonizer の送出レートより速いレートでパケットを送り続けた場合は Stream Harmonizer のバッファがあふれてしまう．

実際には，全送信ホストの CWND 合計値より Stream Harmonizer のバッファサイズが大きければ，バッファがあふれる前に送信ホストがパケット送出を停止する．本稿での実装ではこのメカニズムによりバッファあふれを防止した．より一般性のある方法として，pause パケットを送信ホストに送る方法やパケットを落として送信側の CWND を減らす方法等を今後の課題として検討している．

2.2　その他の最適化

ストリームの速度ばらつき抑制と直接は関係無いが，パケットロスによる性能低下を抑えるため Stream Harmonizer では並列 TCP ストリームに対して送信間隔と送信順序の制御 [5] を行う．

送信間隔制御　並列ストリーム全体としてのパケット送出の時間間隔をできるだけ平均化する．この制御により，短期間のバースト的なパケット送出により中間スイッチのバッファがあふれる事を防ぐ．

各ストリームが流量の制限値一杯までパケットを送った場合に Stream Harmonizer のパケット送出間隔が等間隔になるよう制御する．流量制限により，1 ストリームあたりのパケット送出数が時間 t_s あたり L 個に制限されているとする．Stream Harmonizer はパケットを時間 $t_s/(LN)$　毎に送出する．

送信順序制御　同じストリームのパケットを連続して送出する．これにより連続的にパケットが落ちた場合にパケットロスするストリームの本数を抑える．

図 3 のように，各ストリームのパケットを時間 t_r 毎にまとめて送出する．t_r をストリーム巡回間隔と呼ぶことにする．t_r が長いほど多数のパケットが連続して送出される．t_r が短すぎると，パケットロスするストリームの本数を抑制できない．t_r が長すぎると速いストリームのパケットが多数バースト的に送出されるため，パケットロスの原因となる．

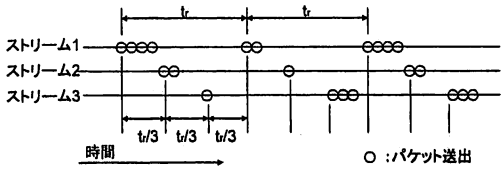

図 3: 同一ストリームのパケット連続送出

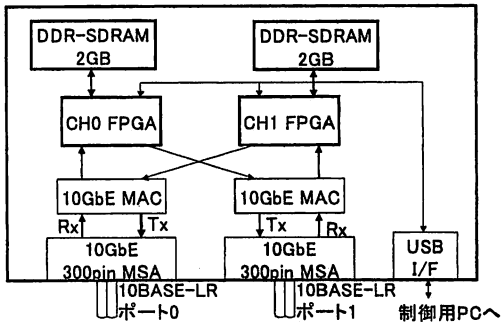

図 4: TGNLE-1 ブロック図

3 実装

10 ギガビット・イーサネット実験装置，TGNLE-1 を用いて Stream Harmonizer の実装を行った．図 4 に TGNLE-1 のブロック図を示す．片方のポートから受信したデータは FPGA に入力され処理された後，もう片方のポートから送信される．

図 5 に実装した Stream Harmonizer のブロック図を示す．入力されたパケットはストリーム毎の FIFO に格納される．FIFO は DDR-SDRAM を用いて実装されている．スケジューラがあるストリームのパケット送信を許可した場合，そのストリームの FIFO からパケットを 1 個読んで送出する．時間 t_s 毎に，パケット送出数を元に各ストリームの流量の上限が計算される．スケジューラは各ストリームの流量上限値を元にパケット送出の順番と時期を決定する．

実装した Stream Harmonizer は扱えるパケット長は最大 8176 オクテット，扱えるストリーム本数は最大 32 である．制御すべき並列 TCP ストリームに属する送信側の IP アドレス範囲および送信側ポート番号範囲をユーザが USB I/F から指定する．ただし，評価では iperf が使うポート番号が毎回変わるため，ポート番号に関わらず全てのストリームを制御対象とするよう変更した．評価対象となる並列 TCP ストリーム以外は Stream Harmonizer に流さないため，評価結果に影響は無い．

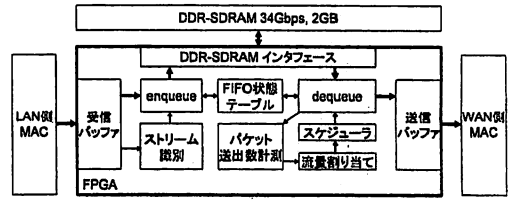

図 5: Stream Harmonizer ブロック図

表 1: FPGA 資源使用率

	使用数	使用率
スライス	9124/23616	54%
18Kbit RAM	34/232	16%

表 1 に Stream Harmonizer の FPGA 資源使用量を示す．Xilinx ISE 6.3.03i を合成に用いた．

4 評価

4.1 評価方法

評価に用いたネットワーク構成を図 6 に示す．ボトルネックが存在する長距離ネットワークをエミュレートする機能を TGNLE-1 に実装した．エミュレータは図 7 に示すネットワークとほぼ等価に振る舞う．ただし，ボトルネックリンク直前でパケットがバッファリングされた場合に生じる遅延はエミュレータでは生じない．このエミュレータを通して並列 TCP ストリームでの通信を行い，通信スループットを測定する．送信ホストから受信ホストへのパケットがボトルネックリンクを通る．ネットワークエミュレータと送信側ホストの間に Stream Harmonizer を入れた場合と入れない場合とで比較を行う．

並列 TCP ストリームを用いたデータ転送時間は最も遅いストリームで決まるので，評価では最も遅いストリームのスループットを議論する．

評価に用いたマシンは CPU が Opteron 248 (2.2GHz, L2 cache 1MB)×2，メモリが DDR-SDRAM PC3200 2GB である．通信には Chelsio 社の 10GbE ネットワークインタフェースカード，T110 を使用する．T110 の TOE 機能は使用しない．使用する OS は Linux 2.6.12 の x86_64 版である．通信には IPv4 を用いる．TCP の輻輳制御アルゴリズムは BIC を用いる．TCP のウィンドウサイズは全ストリームで合計 400MB とする．TCP のスループット測定のため iperf 2.0.2 で 100 秒間の通信を行う．

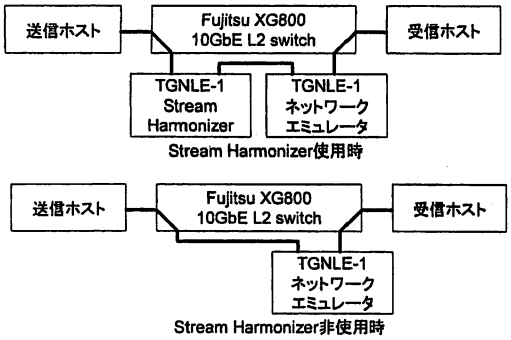

図 6: 評価に用いたネットワーク構成

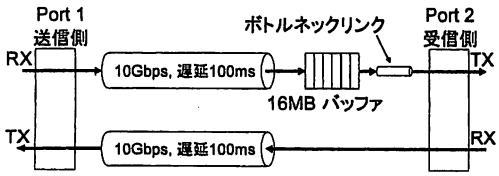

図 7: エミュレートするネットワーク構成

4.2 通信スループット

ネットワークエミュレータの RTT を 200ms，ボトルネックリンク帯域を 3.2Gbps に設定した場合の並列 TCP ストリームの通信速度を表 2 と表 3 に示す．全ストリームの合計速度と，最も遅かったストリームの速度を示す．表に示す値は iperf の通信終了までの平均速度である．表 2 は Stream Harmonizer を使用しない場合，表 3 は Stream Harmonizer を使用した場合の結果である．毎回結果が異なるため，各設定毎に測定を 3 回行った．ストリーム本数に関わらず $t_s = 32$ms，t_r =0.5ms を用いた．この値を用いた場合に最良もしくは最良に近い結果が得られた．α は 2 のべき乗から良い結果を与える値を選んだ．ストリーム本数が 2 の時 $\alpha = 512$，4 と 8 の時は $\alpha = 256$，16 と 32 の時は $\alpha = 128$ を用いた．

Stream Harmonizer を使用した場合，最も遅いストリームの速度が改善されていることが分かる．このことから，Stream Harmonizer によりストリーム間の速度ばらつきを抑制し，並列 TCP ストリーム全体としてのデータ転送時間を短縮可能であることが分かった．全ストリームの合計速度は Stream Harmonizer を使った場合はそうでない場合より低下している．これは，速過ぎるストリームの速度が Stream Harmonizer により抑えられるためである．

図 8 と図 9 に，Stream Harmonizer を使用した場合の各ストリームの速度変化を示した．ストリーム本数は 16 本である．なお，図 1 と図 2 は同じ条件下で Stream Harmonizer を使用しなかった場合の速度変化である．Stream Harmonizer を使用した場合は使用しない場合と比べて最も速いストリームの速度が小さい．また，パケット落ちにより複数のストリームの速度が低下する現象が起きる頻度が低いことが分かる．また，30 秒付近と 60 秒付近において大部分のストリームの速度が抑制されている．そのためネットワーク利用可能帯域が増加し，最も遅いストリームの速度が増加している．

表 2: 通信速度 (Mbps)，Stream Harmonizer 非使用

ストリーム本数	1 回目		2 回目		3 回目	
	合計	最小	合計	最小	合計	最小
2	1180	423	1020	485	1310	400
4	1890	446	2100	254	1660	228
8	1860	164	2000	154	1900	167
16	2170	36.0	2170	23.9	2010	37.8
32	2190	23.8	2320	17.2	2250	16.3

表 3: 通信速度 (Mbps)，Stream Harmonizer 使用

ストリーム本数	1 回目		2 回目		3 回目	
	合計	最小	合計	最小	合計	最小
2	1950	956	1530	732	1530	726
4	1660	288	1740	380	1840	399
8	1680	119	1680	132	1780	177
16	1810	55.9	1890	52.5	1780	78.6
32	1800	35.0	1820	29.9	1780	31.4

5 関連研究

単一ホストでの並列 TCP ストリーム通信の帯域増大を図る方式が提案されている [2][3]．[3] では輻輳ウィンドウ値の増加速度を調節し，他の TCP コネクションとの公平性を保つよう工夫されている．これらの方式は主に単一ホストでのデータ転送速度を上げる事を目的としており，クラスタでの使用を想定している本研究とは目的が異なる．

複数ホストを対象に並列 TCP ストリームの帯域増大を図る方式としてはソフトウェアによる方式 [4] およびハードウェアによる方式 [5] があるが，本研究と異なりエンドホストのソフトウェアを変更する必要がある．

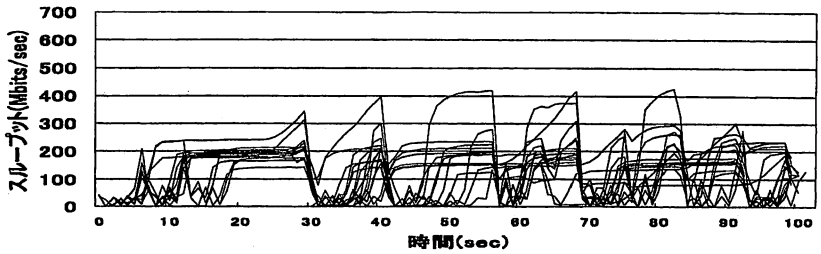

図 8: 並列 TCP ストリームにおける各ストリームのスループット (Stream Harmonizer 使用)

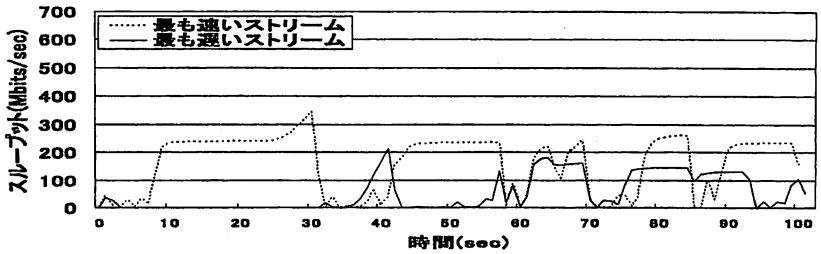

図 9: 並列 TCP ストリームにおけるストリーム間の速度ばらつき (Stream Harmonizer 使用)

6 おわりに

本稿では，並列 TCP ストリームの送信ホストと WAN の中間においてストリーム間の速度ばらつきを抑えることによりデータ転送時間を短縮する機構，Stream Harmonizer を提案した．Stream Harmonizer は各ストリームの速度が全体の平均を大幅に超えないよう制限する．この処理により速すぎるストリームの速度を抑え，遅いストリームを巻き添えにしてパケットロスを起こす頻度を減らす．また，ネットワークの利用可能帯域を増やして遅いストリームの CWND を増加させる．既存方式と異なり，各エンドホストに対する変更は不要である．本研究では，Stream Harmonizer を 10GbE ポートを持つ FPGA 搭載のネットワーク実験装置 TGNLE-1 に実装し，ネットワークエミュレータを用いて評価した．評価の結果，Stream Harmonizer によりストリーム間の速度差を抑えられる事が分かった．

現状では Stream Harmonizer を使用すると，使用しない場合と比べてネットワーク帯域の利用率が低下する．帯域の利用率を高く保ちつつストリーム間の速度ばらつきを抑える事が今後の課題である．

謝辞

本研究は，文部科学省科学技術振興調整費「重要課題解決型研究等の推進–分散共有型研究データ利用基盤の整備」および科学技術振興事業団 CREST による研究領域「情報社会を支える新しい高性能情報処理技術」研究課題「ディペンダブル情報処理基盤」および 21 世紀 COE により実施された．

参考文献

[1] Kei Hiraki, Mary Inaba, Junji Tamatsukuri, Ryutarou Kurusu, Yuukichi Ikuta, Koga Hisashi, Akira Jinzaki, : Data Reservoir: Utilization of Multi-Gigabit Backbone Network for Data Intensive Research, in *Proceedings of the IEEE/ACM Supercomputing(SC2002)* (2002).

[2] Lar Eggert, John Heidemann, Joe Touch, : Effects of Ensemble-TCP, *ACM Computer Communications Review*, Vol. 30, No. 1, pp. 15–29 (2000).

[3] Thomas J. Hacker, Brian D. Noble, Brian D. Athey, : Improving Throughput and Maintaining Fairness using Parallel TCP, in *Proceedings of IEEE Infocom 2004* (2004).

[4] 亀澤 寛之, 中村 誠, 稲葉 真理, 平木 敬, 陣崎 明, 下見 淳一郎, 来栖 竜太郎, 中野 理, 鳥居 健一, 柳沢 敏孝, 生田祐吉 : 長距離・高バンド幅通信における並列 TCP ストリーム間の調停の実現, 先進的計算基盤システムシンポジウム (SACSIS2004), pp. 425–432 (2004 年).

[5] 菅原 豊, 稲葉 真理, 平木 敬 : 細粒度パケット間隔制御の実装と評価, in *IPSJ SIG Technical Reports*, No. 100 in 2005, pp. 85–92 (2005).